

DOCKET: TUC920000080-US1

1. A multi-node network of processors, comprising:

a network;

a plurality of processors coupled in said network, said processors having a minimally operational state, and having a  
5 fully operational state employing a code image, said processors, when in said minimally operational state, requesting said code image from said network; and

a master source coupled in said network, said master source having at least said code image for broadcasting said code image  
10 on said network, said master source, upon receiving said code image request, waiting a predetermined time period, said predetermined time period allowing any additional said processor to reach said minimally operational state, and, upon completion of said predetermined time period, broadcasting said code image  
15 on said network.

2. The multi-node network of processors of Claim 1, wherein said processors, additionally, upon said broadcast of said code image, receive and implement said code image only if said processor is in said minimally operational state.

20 3. The multi-node network of processors of Claim 1, wherein said processors additionally each comprises a non-volatile memory for storing said minimally operational state code.

DOCKET: TUC920000080-US1

4. The multi-node network of processors of Claim 3, wherein said minimally operational state code comprises a boot program which becomes operational upon reboot of said processor.

5. The multi-node network of processors of Claim 3, wherein  
5 said code for said minimally operational state is additionally sufficient to conduct at least a basic system test and provide said code image request.

6. The multi-node network of processors of Claim 1, wherein  
10 said processors additionally comprise a RAM for, upon receiving said code image, storing said code image.

7. The multi-node network of processors of Claim 1, wherein said master source provides one said code image for any said code image request.

8. The multi-node network of processors of Claim 7, wherein  
15 ones of said processors implement different said code images, wherein said one master source code image comprises a combination of said different code images, and wherein said processors additionally select and implement one of said combination of different code images.

DOCKET: TUC920000080-US1

9. The multi-node network of processors of Claim 1, wherein  
said master source comprises a plurality of different said code  
images, wherein said processor requesting said code image  
requests one of said different code images, wherein said master  
5 source broadcasts said requested one of said different code  
images, and wherein said processors additionally determine  
whether said broadcast code image is correct for said processor,  
and select said broadcast code image for implementation if said  
determination determines that said code image is correct for said  
10 processor.

DOCKET: TUC920000080-US1

10. A method for providing a code image for processing nodes of a multi-node network of processors, comprising the steps of:

at least one said processor, comprising a node of said network, in a minimally operational state, requesting said code

5 image from said network;

a master source, upon receiving said code image request, waiting a predetermined time period, said predetermined time period allowing any additional said processor to reach said minimally operational state; and

10 said master source, upon completion of said predetermined time period, broadcasting said code image on said network.

11. The method of Claim 10, additionally comprising the step of, said processors, upon said broadcast of said code image, receiving and implementing said code image only if said processor  
15 is in said minimally operational state.

12. The method of Claim 10, wherein said code of said minimally operational state is stored in a non-volatile memory.

13. The method of Claim 12, wherein said minimally operational state additionally comprises operation of a boot program which  
20 becomes operational upon reboot of said processor.

DOCKET: TUC920000080-US1

14. The method of Claim 12, wherein said minimally operational state of said processor comprises having said non-volatile code additionally sufficient to conduct at least a basic system test and provide said code image request.

5 15. The method of Claim 10, additionally comprising the step of, said processor, upon receiving said code image, storing said code image in RAM.

16. The method of Claim 10, wherein one said code image is provided by said master source for any said code image request.

10 17. The method of Claim 16, wherein ones of said processors implement different said code images, wherein said one code image provided by said master source comprises a combination of said different code images, and wherein said method additionally comprises the step of, said processors selecting and implementing  
15 one of said combination of different code images.

DOCKET: TUC920000080-US1

18. The method of Claim 10, wherein said master source comprises  
a plurality of different said code images, wherein said step of  
said processor requesting said code image comprises requesting  
one of said different code images, wherein said step of said  
5 master source broadcasting said code image comprises broadcasting  
said requested one of said different code images, and wherein  
said method additionally comprises the step of, a receiving said  
processor determining whether said broadcast code image is  
correct for said processor, and selecting said code image for  
10 implementation if said determination step determines that said  
code image is correct for said processor.

DOCKET: TUC920000080-US1

19. For a multi-node network of processors, said network having a plurality of processors coupled in said network, said processors having a minimally operational state, and having a fully operational state employing a code image, said processors,  
5 when in said minimally operational state, requesting said code image from said network, a master source comprising:

a master interface coupled in said network;

a memory storing at least said code image for said processors of said network; and

10 a master processor coupled to said memory and to said master interface, upon receiving said code image request at said master interface, waiting a predetermined time period, said predetermined time period allowing any additional said processor to reach said minimally operational state, and, upon completion  
15 of said predetermined time period, broadcasting said code image stored in said memory, via said master interface, on said network.

20. The master source for said multi-node network of processors of Claim 19, wherein said master processor provides one said code  
20 image for any said code image request.

DOCKET: TUC920000080-US1

21. The master source for said multi-node network of processors of Claim 20, wherein ones of said processors implement different said code images, and wherein said one master source code image stored in said memory and broadcast by said master processor  
5 comprises a combination of said different code images, such that said processors additionally select and implement one of said combination of different code images.

22. The master source of said multi-node network of processors of Claim 19, comprising a plurality of different said code  
10 images, wherein said processor requesting said code image requests one of said different code images, wherein said master source master processor selects from said memory and broadcasts said requested one of said different code images, such that a receiving said processor additionally determines whether said  
15 broadcast code image is correct for said processor, and selects said broadcast code image for implementation if said determination determines that said code image is correct for said processor.



DOCKET: TUC920000080-US1

23. For a multi-node network of processors, said network having a master source coupled in said network, said master source having a code image for broadcasting on said network, said master source, upon receiving said code image request, waiting a  
5 predetermined time period, said predetermined time period allowing any additional processor to reach said minimally operational state, and, upon completion of said predetermined time period, broadcasting said requested code image on said network, a processor comprising:
- 10 a processor interface coupling said processor in said network;
- a non-volatile memory for storing code providing a minimally operational state of said processor;
- a processor memory storing a code image providing a fully  
15 operational state of said processor; and
- a processing unit coupled to said non-volatile memory, said processor memory and said processor interface, when in said minimally operational state provided by said non-volatile memory, requesting said code image from said network, via said processor  
20 interface.

DOCKET: TUC920000080-US1

24. The processor for a multi-node network of processors of Claim 23, wherein said processing unit, additionally, upon said broadcast of said code image by said master source, receives and stores said code image in said processor memory, and implements  
5 said code image, only if said processing unit is in said minimally operational state.

25. The processor for a multi-node network of processors of Claim 23, wherein said minimally operational state code stored in said non-volatile memory comprises a boot program which is  
10 provided to said processing unit and becomes operational upon reboot of said processor.

26. The processor for a multi-node network of processors of Claim 25, wherein said boot program stored in said non-volatile memory storing for said minimally operational state is  
15 additionally sufficient to conduct at least a basic system test and provide said code image request.

27. The processor for a multi-node network of processors of Claim 23, wherein said processor memory comprises a RAM for storing said code image.

DOCKET: TUC920000080-US1

28. The processor for a multi-node network of processors of Claim 23, wherein said master source provides one said code image for any said code image request, said one code image comprising a combination of said different code images, and wherein ones of  
5 said processors implement different said one code images, said processing unit additionally selecting, storing and implementing one of said combination of different code images.

29. The processor of a multi-node network of processors of Claim 23, wherein said master source comprises a plurality of different  
10 said code images, said master source broadcasting said requested one of said different code images, and wherein said processor requesting said code image requests one of said different code images, said processor additionally determines whether said broadcast code image is correct for said processor, and selects  
15 said broadcast code image for implementation if said determination determines that said code image is correct for said processor.

30. A computer program product usable with a programmable computer having computer readable program code embodied therein, said programmable computer comprising a master source coupled in a network, said network having a plurality of processors coupled  
5 in said network, said processors having a minimally operational state, and having a fully operational state employing a code image, said processors, when in said minimally operational state, requesting said code image from said network, said computer program product comprising:

- 10 computer readable program code which causes said master source programmable computer to store at least said code image;  
computer readable program code which causes said master source programmable computer to, upon receiving said code image request, wait a predetermined time period, said predetermined  
15 time period allowing any additional said processor to reach said minimally operational state; and  
computer readable program code which causes said master source programmable computer to, upon completion of said time period, broadcast said stored code image on said network.

20 31. The computer program product of Claim 30, wherein ones of said processors implement different said code images; wherein said stored code image comprises a combination of said different code images; and wherein said computer readable program code

DOCKET: TUC920000080-US1

which causes said master source computer processor to broadcast  
said stored code image, causes said master source computer  
processor to broadcast said code image combination of said  
different code images as one code image, such that said  
5 processors additionally select and implement one of said  
combination of different code images.

32. The computer program product of Claim 30, wherein said  
stored code image comprises a plurality of different said code  
images, wherein said processor requesting said code image  
10 requests one of said different code images, and wherein said  
computer readable program code which causes said master source  
computer processor to broadcast said stored code image, causes  
said master source computer processor to broadcast said requested  
one of said different code images, such that said processors  
15 additionally determine whether said broadcast code image is  
correct for said processor, and select said broadcast code image  
for implementation if said determination determines that said  
code image is correct for said processor.

DOCKET: TUC920000080-US1

33. A method for updating code images for processors of modules of a redundant system, said redundant system comprising at least two sets of redundant said modules, said processors comprising nodes of a multi-node network of processors, said processors  
5 having a minimally operational state and requiring a code image to become fully operational, said processors, when in said minimally operational state, request said code image from said network, said method comprising the steps of:

providing a reboot of at least one said set of redundant  
10 modules, such that said processors of said modules reach said minimally operational state and request said code image from said network;

a master source, upon receiving said code image request, waiting a predetermined time period, said predetermined time  
15 period allowing any additional said processor to reach said minimally operational state;

said master source, upon completion of said predetermined time period, broadcasting said code image on said network, such that said processors requesting said code image become fully  
20 operational;

providing a reboot of a remaining at least one said set of redundant modules, such that said processors of said modules reach said minimally operational state and request said code image from said network;

DOCKET: TUC920000080-US1

said master source, upon receiving said code image request, waiting a predetermined time period, said predetermined time period allowing any additional said processor to reach said minimally operational state; and

- 5        said master source, upon completion of said predetermined time period, broadcasting said code image on said network, such that said processors of said remaining set of redundant modules requesting said code image become fully operational.

DOCKET: TUC920000080-US1

34. A computer program product usable with a programmable computer having computer readable program code embodied therein, said programmable computer comprising a master source coupled in a network, for updating code images for processors of modules of a redundant system, said redundant system comprising at least two sets of redundant said modules, said processors comprising nodes of a multi-node network of processors, said processors having a minimally operational state and requiring a code image to become fully operational, said processors, when in said minimally operational state, request said code image from said network said computer program product comprising:

computer readable program code which causes said master source programmable computer to store at least an update of said code image;

15 computer readable program code which causes said master source programmable computer to provide a reboot of at least one said set of redundant modules, such that said processors of said modules reach said minimally operational state and request said code image from said network;

20 computer readable program code which causes said master source programmable computer to, upon receiving said code image request, wait a predetermined time period, said predetermined time period allowing any additional said processor to reach said minimally operational state;



DOCKET: TUC920000080-US1

computer readable program code which causes said master source programmable computer to, upon completion of said predetermined time period, broadcast said code image on said network, such that said processors requesting said code image  
5 become fully operational;

computer readable program code which causes said master source programmable computer to provide a reboot of a remaining at least one said set of redundant modules, such that said processors of said modules reach said minimally operational state  
10 and request said code image from said network;

computer readable program code which causes said master source programmable computer to, upon receiving said code image request, wait a predetermined time period, said predetermined time period allowing any additional said processor to reach said  
15 minimally operational state; and

computer readable program code which causes said master source programmable computer to, upon completion of said predetermined time period, broadcast said code image on said network, such that said processors of said remaining set of  
20 redundant modules requesting said code image become fully operational.

DOCKET: TUC920000080-US1

35. A multi-node network of processors, comprising:

a network;

a first set of redundant processors coupled in said network, said processors having a minimally operational state, and having  
5 a fully operational state employing a code image, said processors, when in said minimally operational state, requesting said code image from said network;

a second set of redundant processors coupled in said network, said processors having a minimally operational state,  
10 and having a fully operational state employing a code image, said processors, when in said minimally operational state, requesting said code image from said network; and

a master source coupled in said network, said master source having at least said code image for broadcasting said code image  
15 on said network, said first set of redundant processors rebooted to said minimally operational state, said master source, upon receiving said code image request from one of said first set of redundant processors, waiting a predetermined time period, said predetermined time period allowing any additional said processor  
20 to reach said minimally operational state, and, upon completion of said predetermined time period, broadcasting said code image on said network, whereby said first set of redundant processors become fully operational; whereupon said second set of redundant processors are subsequently rebooted to said minimally

DOCKET: TUC920000080-US1

operational state, said master source, upon receiving said code image request from one of said second set of redundant processors, waiting a predetermined time period, said predetermined time period allowing any additional said processor 5 to reach said minimally operational state, and, upon completion of said predetermined time period, broadcasting said code image on said network, whereby said second set of redundant processors become fully operational.